REMARKS/ARGUMENTS

The foregoing amendment and the following arguments are provided to impart precision to the claims, by more particularly pointing out the invention, rather than to avoid prior art.

35 U.S.C. § 102(e) Rejections

The Examiner rejected claims 1-32 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent 6,307,574 (hereinafter "Ashe").

Response to 35 U.S.C. § 102(e) Rejections

Applicant notes that there are similarities in terminology between the present invention and Ashe that appear to indicate likeness between both inventions. However, Applicant respectfully argues that there are significant differences in meaning behind said terminology.

Importantly, it appears that the Examiner is equating the term "multi-level" used in Ashe with the term "multi-layer" used in the present description. However, in Ashe a multi-level graphic file consists of a hierarchical structure with a top level defining the functionality and structure of graphics objects, and a lower level that defines the actual appearance of the objects. Such a hierarchical code organization is disclosed throughout Ashe. See, for example, Figure 4 and Col. 3, lines 7-18.

> (Col. 3, lines 7-18)
>
> In accordance with the present invention, these objectives are achieved by organizing the program code relating to graphical user interface elements, such as menus and control objects, in a multi-level hierarchical structure. At one level of the structure, each different type of menu and control defines a class of objects. The definition of a class includes most, if not all, of the functionality associated with the object of that class. In addition, the class definition can include the overall structure of the object, i.e., the relative positions of different elements which make up the object. The actual appearance of these elements, however, is defined by user selectable software that resides at a lower level of the hierarchy. (emphasis added)

Thus, in Ashe, each level in the hierarchy encapsulates different attribute sets of the GUI elements. Hence, the top levels contain "object classes" that define the functionality and structure of "control objects", whilst the bottom level of the hierarchical code structure defines the appearance of the "elements" that control objects are made up of.

> (See Col. 3, lines 9-17)
>
> At one level of the structure, each different type of menu and control defines a class of objects. The definition of a class includes most, if not all, of the functionality associated with that objects of that class. In addition, the class definition can include the overall structure of the object. The actual appearance of these elements, however, is defined by a user selectable software that resides at a lower level of the hierarchy.

Furthermore, in Ashe, the different hierarchical levels also dictate the degree of user accessibility to editing of the attributes, such that only the lowest level is accessible to a user or developer for editing purposes.

> (See Col. 6, lines 58-67, Col. 7, lines 1-15)
>
> First, the program code at the core class level of the hierarchy is reused for each instance of an object, thereby decreasing the overall size of the operating system, and hence the memory required to store it. Furthermore since the same code is used for each instance of a user interface object, that object will always have the same functionality and general structure, regardless of the particular theme that is chosen for its appearance. The program code at this level of the hierarchy is preferably designed by the operating system developer, who thereby maintains control over the functionality and general structure of the graphical user interface. In contrast, the code at the individual theme level of the structure can be written by different developers. Basically, the code at this level of the structure consists of drawing modules for the individual graphical elements of the menus and control objects. Referring to FIG 2., the code for a scroll bar consists of four drawing modules, one of each of the background, the direction arrows and the thumb. The theme developer can design each of these elements to have any desired appearance. The overall structure of the scroll bar, i.e. the relative positions of the elements, as well as its functionality, remain the same, however, because it is determined by the code at the core level class.
>
> (Emphasis added)

By contrast, in the present invention:

1. A multi-layer graphics file refers to a flat, non-hierarchical list of layers, as disclosed in Figures 4 and 5 and on page 14, lines 3-13 and 23-28. Claim 1 has been amended to state that the multi-layer file is composed of a list of layers. A list of layers is a not hierarchical structure made up of levels. From new claim 1:

> ... the graphic file containing a list of control objects, wherein each control object is in at least one layer...

(emphasis added)

Additionally, unlike Ashe, attributes are not specifically segregated into one level if they refer to functionality and structure, and into another level if they refer to appearance. In the present description, appearance-related attributes and behavior-based attributes may instead be stored in separate files: an “image page” and a “layer list page”, respectively. These files do not have a hierarchical relationship to one another.

See page 14, lines 3-13:

> These picture-related control objects may be embodied in an image page 200 in the graphic file 56 as shown in Figure 4. The control objects 202 depict each control element ad they would substantially appear on the user interface.. Thus, the graphic file, through control objects stored in separate layers, contains the attributes required to display the entire GUI and there is no need to convert the images from a graphic file to an intermediate format.

(emphasis added)

And, see page 14, lines 24-28:

> The control objects that correspond to control element behavior may be recorded in a layer list page 204 in the graphic file 56. One such layer list page 204 is shown in Figure 5. The page has multiple layers 208 of control objects 206 describing some typical behavior-related attributes, where usually each entry represents a single layer having a control object.

(emphasis added)

2. Each control object, which is in at least one layer, is accessible to a user or developer for editing purposes. From new claim 1:

> ...each control object is in at least one layer, dictates at least one attribute of a control element and is editable by a user...

(emphasis added)

Additionally, see page 16, lines 19-21:

> By this direct access configuration, a GUI designer may easily change any aspect of the user interface at any time by altering the graphic file and redrawing individual layers contained therein.

(emphasis added)

And, see Page 15, lines 25-28:

> In order to alter the control elements (attribute) represented by control objects on the layer list page 204, the control object in a layer is reconfigured. The control object in a particular layer is selected and the description of the attribute of interest is changed, e.g. retyped via a keyboard input or menu selection.

(emphasis added)

3. Layers can be grouped together and linked to one another, to share behavior and editing changes. From new claim 4:

> ... the at least one layer is grouped with other layers.

And from new claim 33:

> ... the at least one layer is linked with other layers.

Also, see page 15, lines 28-33 and page 16, lines 1-2:

> For convenience, control objects in multiple layers may optionally be grouped together so that all objects of a particular group will be automatically transferred to the other group members. In addition, members of a group may be further subdivided into subgroups. Another type of connection is a link among layers through link control 220. Linked layers may be placed in the same general location in the graphic file and attribute changes made to any single layer are applied to all associated links.

(emphasis added)

CONCLUSION

Despite similarities in terminology there exist major differences in technique and organization of the code between Ashe and the present description. Therefore, Applicants respectfully submit the present application is in condition for allowance. If the Examiner believes a telephone conference would expedite or assist in the allowance of the present application, the Examiner is invited to call the undersigned at (408) 720-8300.

Extension of Time

Applicants respectfully request a two-month extension of time to respond to the pending Office Action, and a check for the necessary extension fee is enclosed herewith. Authorization is hereby given to charge our Deposit Account No. 02-2666 for any shortage of fees that may be due.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: 7/1, 2005

James C. Sheller, Jr.
Reg. No. 31,195

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300